Dominik Wawszczak
Student ID Number: 440014
Group Number: 1

Parameterized Algorithms
Assignment 3

Problem 1

We will show an fpt-reduction from the Multicolored Clique problem to the Cut with Forbidden Pairs problem. Let (G, k), where $V(G) = (V_1, V_2, \ldots, V_k)$, be an instance of the Multicolored Clique problem. For each $i \in \{1, 2, \ldots, k\}$, let $V_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,n_i}\}$ denote the set of vertices of color i.

We construct a directed graph D as follows:

1. Start with $V(D) = \{s, t\}$ and $E(D) = \emptyset$.
2. For each $i \in \{1, 2, \ldots, k\}$:
   2.1 Add $n_i - 1$ vertices to $V(D)$, denoted as $u_{i,1}, u_{i,2}, \ldots, u_{i,n_i-1}$.
   2.2 For every $v_{i,j} \in V_i$, add an arc $a(v_{i,j})$ to $E(D)$, where $a(v_{i,j}) = (u_{i,j-1}, u_{i,j})$. Here, $u_{i,0} = s$ and $u_{i,n_i} = t$.

Let $\mathcal{S}$ be the set of all pairs of arcs in D whose corresponding vertices are not adjacent in G. Formally:

$$\mathcal{S} \; = \; \{\{a(v_{i_1,j_1}), a(v_{i_2,j_2})\} \; : \; v_{i_1,j_1}, v_{i_2,j_2} \in V(G) \; \wedge \; \{v_{i_1,j_1}, v_{i_2,j_2}\} \notin E(G)\}.$$

The parameter k remains the same as in the original problem. Note that the size of the constructed instance is polynomial in the size of the original instance.

We now prove the following equivalence:

$$(G, k) \in \text{Multicolored Clique} \iff (D, \mathcal{S}, k) \in \text{Cut with Forbidden Pairs}.$$

First, assume there exists a multicolored clique $C = \{v_{1,j_1}, v_{2,j_2}, \ldots, v_{k,j_k}\}$ in G. Define $F = \{a(v_{1,j_1}), a(v_{2,j_2}), \ldots, a(v_{k,j_k})\}$. Since there are exactly k paths from s to t in D, and $a(v_{i,j_i})$ lies on the i-th path for each i, F intersects every path from s to t. Furthermore, for any $\{a_1, a_2\} \in \mathcal{S}$, $|F \cap \{a_1, a_2\}| \leqslant 1$, as otherwise C would not be a clique in G.

Conversely, assume there exists a set $F \subseteq E(D)$ of size k such that F intersects every path from s to t, and $|F \cap \{a_1, a_2\}| \leqslant 1$ for any $\{a_1, a_2\} \in \mathcal{S}$. Let $C = a^{-1}(F)$. Since F intersects every path from s to t, C contains exactly one vertex from each V_i. Additionally, for any two vertices $v_{i_1,j_{i_1}}, v_{i_2,j_{i_2}} \in C$, they are adjacent in G, as otherwise $\{a(v_{i_1,j_{i_1}}), a(v_{i_2,j_{i_2}})\}$ would belong to $\mathcal{S}$. Hence, C is a multicolored clique in G.

From the above, we conclude that the Cut with Forbidden Pairs problem is W[1]-hard when parameterized by k.

By the corollary 14.23 from the Platypus Book, we know that there is no $f(k) \cdot n^{o(k)}$-time algorithm for the Multicolored Clique problem, for any computable function f, assuming the Exponential Time Hypothesis (ETH). By the observation 14.22 from the same book, this implies that no $f(k) \cdot n^{o(k)}$-time algorithm exists for the Cut with Forbidden Pairs problem under ETH, which completes the proof.

Problem 2

Let (D, s, t, k) be an instance of the Double Cut problem and let F be an optimal solution to this problem, meaning one with the smallest possible size. Denote P as the set of all the paths in D from s to t. Define:

$$X = \left\{ a \in F \ : \ \underset{p \in P}{\forall} \ a \text{ is not the last arc on } p \text{ that belongs to } F \right\}$$

and

$$Y = \left\{ a \in F \ : \ \underset{p \in P}{\exists} \ a \text{ is the last arc on } p \text{ that belongs to } F \right\}.$$

With these definitions, $F = X \cup Y$ and $X \cap Y = \emptyset$.

<u>Lemma 1</u> For any $p \in P$, $|p \cap X| \geqslant 1$.

<u>Proof of Lemma 1</u> Suppose, for the sake of contradiction, that there exists a path $p \in P$ such that $p \cap X = \emptyset$. Then, $p \cap F = p \cap Y$. Let a be the first arc on p that belongs to Y. Since $a \in Y$, there exists another path $p' \in P$ where a is the last arc belonging to F. By merging the prefix of p ending with a with the suffix of p' starting after a, we construct another path from s to t that contains only one arc from F. This contradiction completes the proof of Lemma 1.

<u>Lemma 2</u> For some optimal solution F, Y is an important (s, t)-separator in D.

<u>Proof of Lemma 2</u> If there exists a path $p \in P$ that does not pass through Y, then $p \cap F = p \cap X$. Consequently, the last arc on p belonging to F would also be in X, leading to a contradiction. Hence, Y is an (s, t)-separator in D.

Let $R_s(Y)$ denote the set of vertices reachable from s in $D \setminus Y$. Suppose there exists an (s, t)-separator Y' such that $|Y| \geqslant |Y'|$ and $R_s(Y) \subseteq R_s(Y')$, where $R_s(Y')$ is defined analogously. We claim that $F' = X \cup Y'$ is also a valid solution.

If there exists a path $p \in P$ containing at most one arc from F', that arc must belong to Y' because Y' is an (s, t)-separator. By Lemma 1, this arc belongs to X as well. Since F is a solution, p also contains another arc $a = (u, v) \in Y$, where $u \in R_s(Y)$. This implies $u \in R_s(Y')$, meaning there exists a path p' from s to u that avoids Y'. By merging the prefix of p' ending at u with the suffix of p starting at u, we obtain a path from s to t that contains no edges from Y'. This contradicts the assumption that Y' is an (s, t)-separator, proving that F' is another solution of size no greater than F, completing the proof of Lemma 2.

To find an optimal set X for a given Y, we can merge all vertices with outgoing arcs in Y into a single vertex and then run a maximum flow algorithm from s to this new vertex. The correctness of this approach follows from Lemma 1, and its time complexity is polynomial in n, where n is the number of vertices in D.

The final algorithm proceeds as follows:

1. Iterate over all possible important separators Y of size at most k.
2. For each Y, compute the optimal set X.
3. If $|X| + |Y| \leqslant k$, return `true`.
4. If no solution is found, return `false`.

The time complexity of this algorithm is $4^k \cdot n^{\mathcal{O}(1)}$, as important separators can be enumerated in time $4^k \cdot n^{\mathcal{O}(1)}$. This is fixed-parameter tractable when parameterized by k, concluding the proof.

Problem 3

We will use the representative sets technique. Define

$$\begin{aligned}\mathcal{P}^p_{u\to v} \;=\; \Big\{V(P)\ :\ & P \text{ consists of } \left\lfloor \tfrac{p-1}{5} \right\rfloor \text{ vertex-disjoint cycles of length 5}\\ & \text{and a path of length } (p-1) \bmod 5 \text{ from } u \text{ to } v\Big\},\end{aligned}$$

for every $u, v \in V(G)$. Let $\widehat{\mathcal{P}}^p_{u\to v} \subseteq^{5k-p}_{\text{rep}} \mathcal{P}^p_{u\to v}$. We will calculate $\widehat{\mathcal{P}}^p_{u\to v}$ for each $p \in \{1, 2, \ldots, 5k\}$, maintaining the invariant $\left|\widehat{\mathcal{P}}^p_{u\to v}\right| \leqslant \binom{5k}{p}$.

For $p = 1$, we have $\widehat{\mathcal{P}}^1_{u\to u} = \{\{u\}\}$, for every $u \in V(G)$. Assume that $\widehat{\mathcal{P}}^{p-1}_{u\to v}$ has already been computed. If $(p-1) \bmod 5 \neq 0$, let

$$\widetilde{\mathcal{P}}^p_{u\to v} \;=\; \bigcup_{\{w,v\}\in E(G)} \Big\{P \cup \{v\}\ :\ P \in \widehat{\mathcal{P}}^{p-1}_{u\to w}\ \wedge\ v \notin P\Big\},$$

for every $u, v \in V(G)$. Otherwise, let

$$\widetilde{\mathcal{P}}^p_{u\to u} \;=\; \bigcup_{\{v,w\}\in E(G)} \Big\{P \cup \{u\}\ :\ P \in \widehat{\mathcal{P}}^{p-1}_{v\to w}\ \wedge\ u \notin P\Big\},$$

for every $u \in V(G)$. Here we only consider vertex pairs connected by an edge to close the cycle. Finally, let

$$\widehat{\mathcal{P}}^p_{u\to v} \;=\; \operatorname{trim}\Big(\widetilde{\mathcal{P}}^p_{u\to v}\Big),$$

for every $u, v \in V(G)$. The trim operation applies the FLS (Fomin, Lokshtanov, Saurabh) algorithm, ensuring the invariant is maintained.

The final answer is

$$\exists_{\{u,v\}\in E(G)}\ \widehat{\mathcal{P}}^{5k}_{u\to v} \;\neq\; \emptyset,$$

since the last cycle must be closed by an edge.

The size of $\widetilde{\mathcal{P}}^p_{u\to v}$ is at most $n^2 \cdot \binom{5k}{p}$ due to the invariant. Therefore, the time needed to compute $\widehat{\mathcal{P}}^{5k}_{u\to v}$ is bounded by $(2^{5\omega})^k \cdot n^{\mathcal{O}(1)}$, where ω is the matrix multiplication constant. Given that there are $5k$ iterations, each requiring this step n^2 times, the total time complexity is $(2^{5\omega})^k \cdot n^{\mathcal{O}(1)}$. This completes the proof, as $2^\omega \approx 2^{2.371552} \approx 5.174975$, while $2e \approx 5.436564$.

Problem 4

Without loss of generality, assume the decomposition is nice. For any subtree H of the decomposition and its boundary ∂H, let $f : V(\partial H) \to \{\text{blue}, \text{red}\}$. Define $\text{dp}_H(f)$ as the maximum number of edges in H with endpoints of different colors, for some coloring $g : V(H) \to \{\text{blue}, \text{red}\}$ such that for every $u \in V(\partial H)$, $g(u) = f(u)$.

To compute dp_H, we consider the following cases:

1. $H = \emptyset$

   In this case, $\text{dp}_H(\text{empty function}) = 0$.

2. $H = \text{introduceVertex}(H', u)$

   Here, $\text{dp}_H(f) = \text{dp}_{H'}(f')$, where $f = f'[u \to \text{color}]$, and color is either blue or red. We use the notation $g[p \to q]$ to denote the function h defined by:

$$h(x) \;=\; \begin{cases} q, & \text{if } x = p,\\ g(x), & \text{otherwise.}\end{cases}$$

3. $H = \text{introduceEdge}(H', u, v)$

   In this case, $\text{dp}_H(f) = \text{dp}_{H'}(f) + [f(u) \neq f(v)]$, where $[P]$ denotes the Iverson bracket, i.e., $[P] = 1$ if P is true and $[P] = 0$ otherwise.

4. $H = \text{forgetVertex}(H', u)$

   Here, $\text{dp}_H(f) = \max(\{\text{dp}_{H'}(f[u \to \text{blue}]), \text{dp}_{H'}(f[u \to \text{red}])\})$.

5. $H = \text{merge}(H', H'')$

   In this case $\partial H = \partial H' = \partial H''$, and we set

   $$\text{dp}_H(f) \;=\; \text{dp}_{H'}(f) + \text{dp}_{H''}(f) - |\{\{u, v\} \;:\; \{u, v\} \in E(\partial H) \;\wedge\; f(u) \neq f(v)\}|.$$

   We subtract the number of edges in ∂H with endpoints of different colors to avoid double counting.

The answer to the MAX CUT problem is $\text{dp}_T(\text{empty function})$, where T is the full decomposition. The time complexity of this algorithm is $2^t \cdot n^{\mathcal{O}(1)}$.

Now, we prove that under the Strong Exponential Time Hypothesis (SETH), no algorithm can run in time $2^{t-\varepsilon} \cdot n^{\mathcal{O}(1)}$ for any $\varepsilon > 0$. Given an instance φ of the SAT problem with n variables, we construct a graph G as follows:

1. Create a special vertex r.

2. For each variable x_i, add a vertex u_i to G. Define $u(l)$ as the vertex corresponding to literal l.

3. For each clause $c \in \varphi$, let $c = (l_{c,1} \vee l_{c,2} \vee \ldots \vee l_{c,|c|})$, where $l_{c,j} \in \bigcup_{i \in \{1,2,\ldots,n\}} \{x_i, \neg x_i\}$, for each $j \in \{1, \ldots, |c|\}$. Without loss of generality, assume $|c| \leqslant n$, as otherwise, some two literals would correspond to the same variable. Additionally, assume the first k_c literals correspond to normal variables, while the remaining $|c| - k_c$ are negations.

   3.1 Create $2n+2$ vertices: $v_{c,1}, v_{c,2}, \ldots, v_{c,2n+2}$. Arrange them in a cycle $C_c = (r, v_{c,1}, v_{c,2}, \ldots, v_{c,2n+2})$, adding $2n + 1$ edges between every pair of consecutive vertices. This results in a non-simple graph, but we will later address this issue.

   3.2 For each $j \in \{1, 2, \ldots, k_c\}$, add edges $\{v_{c,2j-1}, u(l_{c,j})\}$ and $\{v_{c,2j}, u(l_{c,j})\}$.

   3.3 For each $j \in \{k_c + 1, k_c + 2, \ldots, |c|\}$, add edges $\{v_{c,2j}, u(l_{c,j})\}$ and $\{v_{c,2j+1}, u(l_{c,j})\}$.

   Note that by this construction, every vertex in C_c, except for r, has at most one neighbor outside the cycle.

Let

$$M \;=\; ((2n + 1) \cdot (n + 1) + 1) \cdot m + \sum_{c \in \varphi} |c|,$$

where m is the number of clauses in φ. We claim that φ is satisfiable if and only if the max cut in G is at least M.

First, assume that φ is satisfiable. We assign r the color blue. For each variable x_i, we color u_i blue if x_i is true, and red otherwise. For every clause c, let $l_{c,j}$ be the first true literal in c. Let $v_{c,x}$ and $v_{c,x+1}$ be the two neighbors of $u(l_{c,j})$ in C_c. For all $y \in \{1, 2, \ldots, x\}$, we color $v_{c,y}$ blue if y is even, and red otherwise. For all $y \in \{x + 1, x + 2, \ldots, 2n + 2\}$, we color $v_{c,y}$ red if y is even, and blue otherwise. Note that $v_{c,x}$ and $v_{c,x+1}$ share the same color, which differs from the color of $u(l_{c,j})$, while the remaining vertices in C_c alternate in color. Thus, the number of edges with endpoints of different colors in C_c is exactly $(2n+1) \cdot (n+1) + 1 + |c|$, since the neighbors of every literal vertex other than $u(l_{c,j})$ have different colors. Therefore, the total number of edges with endpoints of different colors is M.

Conversely, assume that the max cut in G is at least M. Without loss of generality, assume the color of r is blue. Since for every clause $c \in \varphi$, the cycle C_c has length $2n+3$, there can be at most $n+1$ pairs of consecutive vertices with different colors. If there exists a clause $c \in \varphi$ with fewer than $n+1$ such pairs, the number of edges with endpoints of different colors in C_c would be at most $(2n+1) \cdot n + 2|c|$, which is strictly less than $(2n+1) \cdot (n+1)$. Therefore, for each $c \in \varphi$, there must be exactly one pair of consecutive vertices in C_c with the same color. Moreover, for any clause $c \in \varphi$, the contribution of C_c to the cut is at most $(2n+1) \cdot (n+1) + 1 + |c|$, implying that it is exactly $(2n+1) \cdot (n+1) + 1 + |c|$. Thus, there exists a literal $l_{c,j}$ such that $u(l_{c,j})$ has a different color from both its neighbors in C_c. This is the literal that satisfies the clause c, concluding the proof of the equivalence.

To remove multi-edges, replace each with a path of length 3. It is easy to verify that the equivalence still holds.

Since the size of G is polynomial in the size of φ, it remains to calculate the bound on the treewidth of G. The treewidth of each path of length 3 is 1. We can place all the decompositions of these paths next to each other, then insert the vertices from C_c that each path connects, resulting in a tree decomposition of treewidth 3. Next, we add r and the vertices corresponding to the variables to each bag, which gives a tree decomposition of G with treewidth $n+4$.

Therefore, an algorithm solving Max Cut in time $2^{t-\varepsilon} \cdot n^{\mathcal{O}(1)}$ for some $\varepsilon > 0$ contradicts SETH, which completes the proof.